WILMERHALE

FACSIMILE

+1 617 526 6105 (t)
+1 617 526 5000 (f)
ronald.demsher@wilmerhale.com

Date March 15, 2007

To C. Chow, Examiner
USPTO

Fax 571-273-9393
Tel

cc

From Ronald R. Demsher

Pages (including cover) 4

Re

Examiner Chow,

To follow are the proposed amendments we discussed.

Best regards,

Ronald R. Demsher

PTO Reg. No.: 42,478

Docket No.: 1801270.00124US1
(PATENT)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Jason Souloglou et al.	Confirmation No.:	5417
Application No.:	09/827,971	Art Unit:	2192
Filed:	April 6, 2001	Examiner:	C. C. Chow
Title:	PROGRAM CODE CONVERSION		

MS Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

PROPOSED AMENDMENT IN RESPONSE TO EXAMINERS REQUEST

Dear Sir:

INTRODUCTORY COMMENTS

In response to the Examiner's request, please amend the above-identified U.S. patent application as follows:

Amendments to the Claims are reflected in the listing of claims which begins on page 2 of this paper.

AMENDMENTS TO THE CLAIMS

This listing of the claims will replace all prior versions, and listings, of claims in the application:

1. (Previously presented) A method of generating an intermediate representation of a register-based program code that refers to a set of registers, the method comprising the computer implemented steps of:

generating a plurality of register objects each representing a respective one of said registers as referenced by the program code;

generating one or more expression objects each representing a respective operator or operand of said program code as that element arises in the program code; and

forming a network of said register objects and expression objects with each said expression object being referenced by one or more of said register objects to which it relates either directly, or indirectly via references from other of said expression objects.

2. (Previously presented) The method according to claim 1, wherein said program code is expressed in terms of an instruction set of a subject processor.

3. (Previously presented) The method according to claim 2, wherein said register objects represent the set of registers of said subject processor.

4. (Previously presented) The method according to claim 1, further comprising the step of dividing said-program code into a plurality of basic blocks each having only one effective entry point instruction and one effective exit point instruction, and performing said generating steps sequentially block-by-block with respect to said plurality of basic blocks.

5. (Previously presented) The method according to claim 1, wherein at least some said expression objects feed into more than one said register objects.

6. (Previously presented) The method according to claim 1, wherein said expression objects are not duplicated.

7. (Previously presented) The method according to claim 1, wherein a single said expression object is generated for a given operator or operand of said program code, and each said expression object is referenced by all said register objects to which it relates.

8. (Previously presented) The method according to claim 1, further comprising the step of eliminating one or more said register objects or a said expression objects if they are identified as being redundant or unnecessary.

9. (Previously presented) The method according to claim 8, further comprising the step of identifying a redundant or unnecessary said register object or said expression object by maintaining an ongoing count of references being made to that object as the network of register and expression objects is constructed in said intermediate representation.

10. (Previously presented) The method according to claim 9, wherein for each expression object a count is maintained of the number of references to that expression object from other expression objects or from register objects, the count associated with a particular expression object being adjusted each time a reference to that expression object is made or removed.

11. (Previously presented) The method according to claim 10, wherein an expression object and all references from that expression object are eliminated when said count for that expression object is zero.

12. (Previously presented) The method of claim 1, further comprising the step of translating the program code written for execution by a processor of a first type so that the program code may be executed by a processor of a second type, using the generated intermediate representation.

13. (Previously presented) The method of claim 12, wherein said translating step is performed dynamically as the program code is run.

14. (Previously presented) The method of claim 1, further comprising the step of optimising the program code by optimising the generated intermediate representation.

15. (Previously presented) The method of claim 14, wherein said optimising step is used to optimise the program code written for execution by a processor of a first type so that the program code may be executed more efficiently by that processor.

16. (Previously presented) A method for generating an intermediate representation of a register-based program code written for running on a programmable machine having a set of registers, said method comprising:

(i) generating a plurality of register objects in the intermediate representation, each said register object representing a respective one of the set of registers as referenced by the program code; and

(ii) generating a plurality of expression objects in the intermediate representation, said expression objects representing fixed values and/or relationships between said fixed values and said registers according to said program code;

wherein said register objects and said expression objects are organised into a branched tree-like network having all register objects at the lowest basic root or tree-trunk level of the network with ~~no~~ none of said register objects feeding into any other of said register objects.

17. (Currently amended) A system for generating an intermediate representation of a register-based program code which refers to a set of registers of a processor, comprising:

means for generating a plurality of register objects each representing a respective one of said registers as referenced by the program code;

means for generating one or more expression objects each representing a respective operator or operand of said program code as that element arises in the program code; and

forming a network of said register objects and expression objects with each said expression object being referenced by one or more of said register objects to which it relates either directly, or indirectly via references from other of said expression objects.

18. (Previously presented) A system for generating an intermediate representation of a register-based program code written for running on a programmable machine having a set of registers, the system comprising:

means for generating a plurality of register objects, each said register object representing a respective one of the set of registers as referenced by the program code; and

means for generating a plurality of expression objects representing fixed values and/or relationships between said fixed values and said said registers according to said program code;

wherein said register objects and said expression objects are organised into a branched tree-like network having all of said register objects at the lowest basic root or tree-trunk level of the network with none of said register objects feeding into any other of said register objects.

In view of the above amendment, applicant believes the pending application is in condition for allowance. No other fees are believed to be due in connection with the filing of this response, however the Commissioner is authorized to debit Deposit Account No. 08-0219 for any required fee necessary to maintain the pendency of this application.

Respectfully submitted,

Dated:

Ronald R. Demsher
Registration No.: 42,478
Attorney for Applicant(s)

Wilmer Cutler Pickering Hale and Dorr LLP
60 State Street
Boston, Massachusetts 02109
(617) 526-6000 (telephone)
(617) 526-5000 (facsimile)